

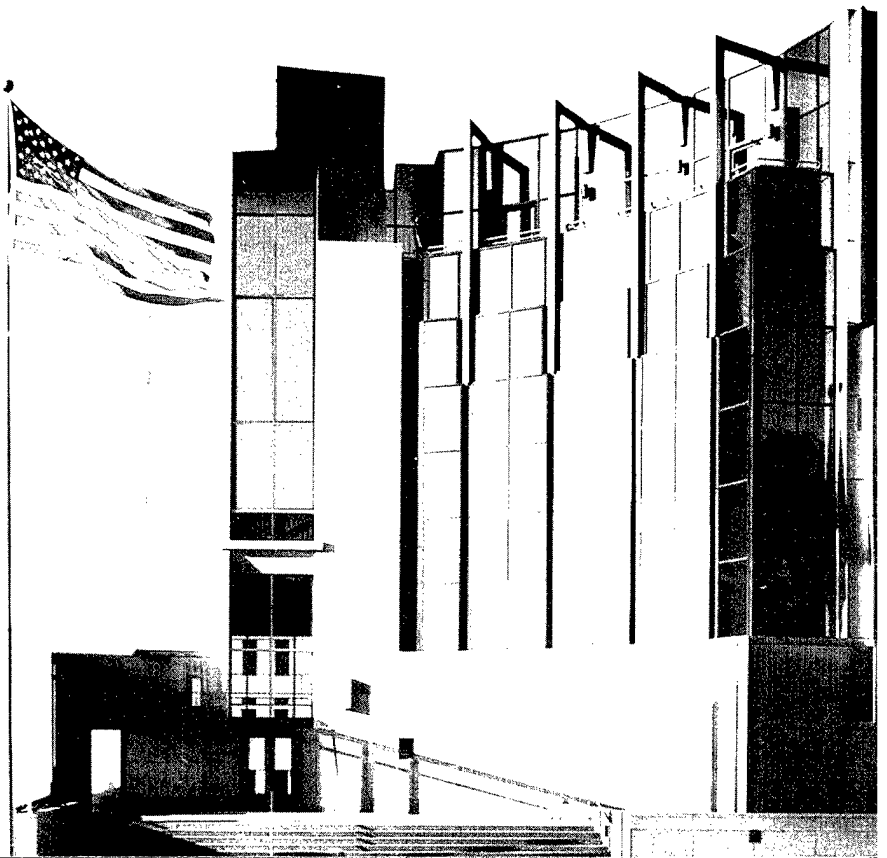
Reeducation to Expand the Software Engineering Workforce: Successful Industry/University Collaborations

Heidi J. C. Ellis
Ana Moreno
Nancy R. Mead
Stephen B. Seidman

July 2002

SPECIAL REPORT
CMU/SEI-2002-SR-001

20020919 009



Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of "Don't ask, don't tell, don't pursue" excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Obtain general information about Carnegie Mellon University by calling (412) 268-2000.



Reeducation to Expand the Software Engineering Workforce: Successful Industry/University Collaborations

CMU/SEI-2002-SR-001

Heidi J. C. Ellis
Ana Moreno
Nancy R. Mead
Stephen B. Seidman

July 2002

Networked Systems Survivability Program

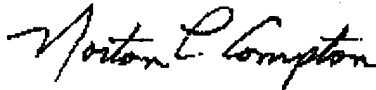
Unlimited distribution subject to the copyright.

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER



Norton L. Compton, Lt Col, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2002 by Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number F19628-00-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (<http://www.sei.cmu.edu/publications/pubweb.html>).

Table of Contents

Abstract.....	vii
1 Introduction	1
1.1 Background.....	1
1.2 The Working Group on Software Engineering Education and Training	2
2 Investigation Process	3
2.1 Focus of the Investigation.....	3
2.2 Our Approach.....	3
3 Successful Collaboration Construction and Execution	7
3.1 Common Characteristics	7
3.2 Dissimilar Characteristics	8
3.3 Benefits	9
3.4 Industry Viewpoint.....	10
4 Evaluating Knowledge/Skill Transfer	13
4.1 Alumni Survey Description	13
4.2 Alumni Survey Results	15
4.2.1 Results of the Content Measurement.....	15
4.2.2 Results of the Subjective Assessment	19
5 Conclusion.....	23
References.....	25

List of Figures

Figure 1: Process Used to Study I/U Reeducation Collaborations4

Figure 2: Weighted Values of Emphasis, Usefulness, and Sufficiency of
Knowledge for Each Subject 15

Figure 3: Graphical Representation of Attendants' Assessment of the Program
Effect.....20

List of Tables

Table 1:	Subject Assessment Criteria Concerning Emphasis, Usefulness, and Alumni's Knowledge	14
Table 2:	Current Alumni Job Types	15
Table 3:	Percentage of Alumni Who Evaluated Each Statement	19

Abstract

Software produced worldwide is growing at a phenomenal rate as software is used in such diverse products as automobiles, homes, and airplanes. In addition, the increasingly global business climate and expanding emphasis on distributed computing have accelerated the need for business software. However, there is currently an inadequate number of software engineers to produce and maintain software to meet this demand. One possible solution to correcting this shortfall is reeducating existing non-software engineering employees to become software engineers. For the past two years, the Industry/University (I/U) subgroup of the Working Group on Software Engineering Education and Training has been investigating active collaborations between companies and universities in which non-software professionals and practitioners who lack formal software education are reeducated to become software engineers. This paper reports on the I/U subgroup's findings by describing their approach to the investigation, the factors involved in successful collaboration construction and execution, and alumni views of the knowledge and skills transferred by the collaboration.

1 Introduction

1.1 Background

In the IDC report titled *Worldwide Software Market Forecast Summary, 2001-2005* [Heiman 01], Heiman et al. have projected a growth rate of 11.8% for companies in the market of developing and deploying software applications in the year 2002. Although this figure is reduced from the 15% rate originally projected before September 11, 2001, it still indicates rapid growth in the development and use of software. However, the number of software professionals available to develop and maintain this large body of software is limited and is not growing at a rate equal to the growth rate of software. The U.S. Department of Labor, Bureau of Labor Statistics projects that jobs in the software-centric categories of “engineering, natural science, and computer and information systems managers” and “computer systems analysts, engineers, and scientists” will be two of the highest growth job areas for the next several years [Kellinson 00]. The lack of software professionals is also occurring in countries such as Canada, India, and most countries in Europe.

Educational institutions are expanding efforts to increase the number of software engineering graduates [Modesitt 01], but the rate of production is not keeping up with demand. The re-education of existing non-software personnel to fill software engineering positions is one logical approach for meeting this shortfall. Indeed, some companies have entered into cooperative partnerships with academic institutions to retrain employees with domain knowledge in order to retain this valuable resource [Frailey 98]. Bertrand Meyer [Meyer 01] indicates the need for software engineering education, as opposed to creating programmers, and various other efforts have identified the need for incorporating a strong industry perspective into software engineering programs [Frezza 99, Hazen 97, Mingas 99, Wholin 99]. Thus, the construction of a collaborative relationship between industry and academic partners to reeducate a portion of the existing non-software workforce seems to be a logical and natural solution to providing a larger software engineering workforce.

This paper presents the findings of recent research into collaborative reeducation efforts between industry and academic institutions to produce qualified software engineering professionals. The research is being performed by the Industry/University (I/U) subgroup of the Working Group on Software Engineering Education and Training. The current focus of our investigation is on active collaborations that exist between companies and universities for converting non-software professionals into software engineers in order to provide larger

numbers of trained software engineers. In the following section, we provide background on the group, including the goals of our work.

1.2 The Working Group on Software Engineering Education and Training

The Working Group on Software Engineering Education and Training (WGSEET) (<http://www.sei.cmu.edu/collaborating/ed/workgroup-ed.html>) was formed in 1995 with the mission of improving the state of software engineering education and training. The WGSEET is an ad hoc group of approximately 80 international professionals from academia, industry, and government with the goal of promoting software engineering education in all academic levels. The focus of the group is on the education and professional development of software practitioners through degree programs, continuing education, on-the-job training, and other educational mechanisms. WGSEET activities include the investigation of issues, proposal of solutions, publication of state-of-the-practice information, and publication of working group activities and results. Current areas of study include best practices in industry/university collaboration and guidelines for software engineering curricula.

The I/U subgroup is a subset of WGSEET members whose focus is to explore and foster collaborations between academic institutions and industry [Mead 99]. Currently, the I/U group is investigating collaborations in which non-software professionals and practitioners without formal software education are reeducated to become software engineers. By studying successful collaborations, we hope to identify the characteristics of a successful joint venture for retraining the future software workforce. We also hope to foster further collaborations and provide guidelines to both universities and companies interested in constructing a collaborative program for reeducating employees to become software engineers. Another goal is to provide feedback on which areas of software engineering knowledge and skills are most easily transferred to the student practitioners and applied in the workplace, in order to understand how well the student practitioners are able to internalize the material and apply it on the job.

The remainder of this paper is organized as follows: Section 2 describes our approach to the research by explaining our methodology and research framework. Section 3 discusses the construction and execution of the successful collaborations, and Section 4 describes the transfer of knowledge and skills to the student practitioners. Section 5 provides some general conclusions and outlines directions for future work.

2 Investigation Process

2.1 Focus of the Investigation

The motivation for the investigation of I/U reeducation collaborations was a talk given to the WGSEET by Dennis Frailey [Frailey 98] in which he described a collaborative software engineering reeducation program between Raytheon and several Dallas area universities. Using Beckman's definition [Beckman 99] of a collaboration as "a formal, joint effort by a university (or universities) and a business or government organization(s), where each party provides specified products and services to achieve common goals," we decided to focus on cooperative programs between academic institutions and industry partners to retrain non-software professionals to become software engineers. As the level of skills and knowledge necessary for an individual to be a proficient software engineer requires a significant effort to obtain, we decided to center our investigation on education programs that had a wider effect than several-day or week-long just-in-time training. When we started our work in March 2000, we identified nine collaborative programs between industry and academic institutions. By February 2001, we had centered our investigation on five ongoing collaborations whose durations ranged from eight months to two years and resulted in students who obtained credit towards undergraduate courses to students who received master's degrees. In the following sections we describe our approach to investigating I/U reeducation collaborations.

2.2 Our Approach

The major steps taken in our investigation and their organization are shown in Figure 1. Our approach was crafted as a result of several meetings of the I/U subgroup and included input from many I/U subgroup members.

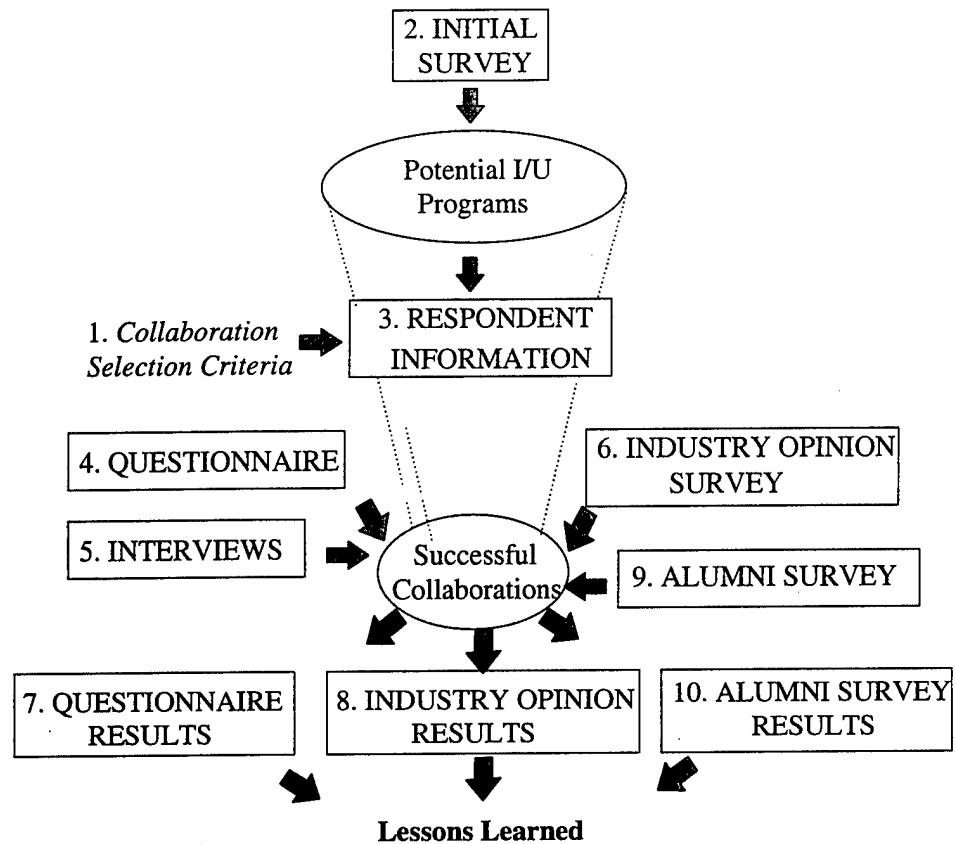


Figure 1: Process Used to Study I/U Reeducation Collaborations

Our investigation process had 10 steps:

Step 1 – Determine Collaboration Selection Criteria: We began our efforts in March 2000 by identifying a set of criteria, which we used to select candidate collaborations. We defined four main selection criteria, including collaborations in which students have no formal software engineering background, collaborations that have been in existence for at least a year, collaborations that are currently in the reeducation process, and collaborations in which ongoing interaction occurs between the university and the industry.

Step 2 – Conduct Initial Survey: Based on our selection criteria, we developed a short survey that we used to identify candidate collaborations. The survey contained nine questions intended to highlight the collaboration selection criteria and to elicit some general information about respondent background, number of students in the program, and future plans for the program. Between April and June 2000, we distributed this survey to a variety of software

engineering forums, including the Forum for Advancing Software Engineering Education (FASE).

Step 3 – Filter Respondent Information: The next step in our process was to apply the selection criteria to the results of the initial survey to determine collaborations. This was carried out in December 2000 and resulted in a small pool of nine successful collaborations, six located in the United States and three located in Europe. Of these collaborative programs, seven resulted in students obtaining a master's degree, one resulted in students obtaining a bachelor's degree, and in one collaboration students received a certificate. Details of the results are described by Ellis [Ellis 01b].

Step 4 – Administer Questionnaire: In order to elicit more detailed information about the collaborations, an in-depth questionnaire was constructed in November 2000. Questions focused on the organizational and participation factors that may have contributed to the success of the collaboration, rather than the content of the programs themselves. The following major areas were covered:

- general background information
- program inputs, such as admission criteria
- program outcomes, such as participant success
- program content
- program format, such as delivery methods
- program budget and management
- overall benefits

Step 5 – Conduct Interviews: Our next step, carried out during January and February 2001, was to administer the questionnaire to the respondents of the initial surveys involved in the identified collaborations. The questionnaire was completed through a two-part process. We first emailed the questionnaire to the collaboration contact person identified in the initial survey, and that person filled out the simple portions of the form (e.g., checkboxes). We then set up a phone interview, in which one member of the I/U group interviewed the contact person to elicit answers to the more detailed questions (e.g., benefits and lessons learned).

Step 6 – Create Industry Opinion Survey: Upon receiving responses from the questionnaires and interviews, we realized that most of the responses were coming from the academic partner in the collaboration, providing a somewhat academically biased viewpoint. In order to elicit industry views, we constructed a short survey during February 2001. The first question in the industry opinion survey asked respondents to rate their satisfaction with 14 aspects of the program, including development process, pace, format, and content, on a three-level scale (dissatisfied, satisfied, very satisfied). The remaining five questions elicited information

about the benefits received, lessons learned, and improvements that could be made to the program.

Step 7 – Evaluate Questionnaire Results: During the month of February 2001, we tabulated the results of both the questionnaire and the industry opinion survey to determine characteristics of successful industry/university collaborations. Ellis et al. provide the initial results of the questionnaire [Ellis 01a]; an overview of features of successful collaborations is included in Section 3.

Step 8 – Incorporate Industry Opinion Results: The industry opinion survey was administered to the industry partners participating in the collaborations in April 2001, and during May 2001 the results were tabulated. Industry opinions on successful collaboration development are included in Section 3; details are provided by Ellis et al. [Ellis 02].

Step 9 – Conduct Alumni Survey: During October 2001, the I/U subgroup decided to complete our study on I/U reeducation collaborations by investigating how effective the reeducation collaborations are in transferring technology and skills to people with non-software engineering backgrounds. As part of this effort, we decided to elicit information from participants in the collaboration, as well as from managers responsible for overseeing graduates of the collaborations. Two surveys were constructed. An alumni survey asked alumni opinions of the breadth of software engineering knowledge gained from the collaboration and the usefulness of that knowledge. Alumni survey details are provided in Section 4. A management survey was constructed that asked similar questions of the managers of the alumni; however, we received only two responses to this survey and the responses were very neutral in nature. Therefore we could not draw any conclusions about management opinions from these two surveys.

Step 10 – Interpret Alumni Survey Results: In February 2002, the I/U subgroup evaluated the results of the alumni survey. While we lacked sufficient responses to obtain statistically significant results, we were able to draw some broad inferences. In general, students received a wide exposure to a range of software engineering technologies, students were satisfied with the knowledge and skills that they obtained, and students would recommend the collaborative programs to others. Section 4 provides details on these results.

3 Successful Collaboration Construction and Execution

Based on our observations of the ongoing collaborations over the course of two years, we have identified characteristics of successful collaborative efforts to reeducate software engineers. While all of the identified features may not be required for the creation of a successful collaboration, we do feel that most of these factors are required. In addition, we have identified several benefits to the university and industry partners participating in the collaborations. In this section, we discuss the characteristics that successful collaborative efforts share and identify the dissimilar characteristics. We describe benefits accrued by the partners in the collaborations and provide insight into the industry opinions on the success of the collaborative efforts. The analysis presented in this section is based on the evaluation of the results of the detailed questionnaires and industry opinion surveys.

3.1 Common Characteristics

Based on our observations, it appears that successful collaborations share factors related to industry participation. One possible conclusion that could be drawn from industry impact is that significant industry participation and input is required for collaborations to succeed. Common characteristics of the collaborations we studied include the following:

Industrial Initiative: Most of the collaborations resulted from the industry partner(s) recognizing the need for more highly qualified software engineers and contacting the university partner, an institution with which they typically had a prior relationship. This relationship appears to be more important than the universities' experience with collaborative programs, as the industry partners selected their university partners with little regard to whether the academic institution had any previous software engineering reeducation programs. The implications of this disregard for experience with collaborative programs is twofold: (1) industry realizes the potential of the academic institution for conveying valuable software engineering; and (2) there is a considerable incentive for universities considering running such training programs.

Strong Industrial Partners: The industry partners in the collaborations are either sizeable companies or collections of companies that have significant financial resources. In all cases, the company partner pays, either directly or indirectly, for student enrollment. One common

feature of the industry partners is that almost all of them are either government agencies or private companies that handle a large number of government contracts.

Practically Oriented Programs: One main characteristic related to the content of the collaborative programs is that they all are practically oriented, focusing on the application of knowledge and skills. In fact, several programs orient assignments and case studies to the specific domain in which the industry partners operate to maximize application of student knowledge.

Audience of Software Practitioners: All of the students who participated in the collaborative programs are employed in some aspect of the software field. Student experience in software development varies from 2 to over 10 years. Many students took part-time courses while working full time. This student background of software practitioners impacts the educational experience in two ways. First, industry partners must be conscious of the effort expended by their employees in the collaborative program. Similarly, the academic partner must also factor in student workload when defining the volume of homework assigned to students.

Limited Number of Students: The successful collaborations that we examined all had between 10 and 25 students per session. In some cases the university partner established the number of students per offering of the program. The decision was typically based on pedagogical reasons, such as the desire to have a group small enough for substantial interaction yet large enough to motivate interaction. In other collaborations, the industry partner defined class size limits typically based on financial and resource considerations.

3.2 Dissimilar Characteristics

In addition to the commonalities described above, we also identified several differences between the programs. These differences were mainly related to program content, admission procedure, and financial models and did not appear to adversely affect the success of the collaborations. We describe the three main differences below.

Content of the Program: While we did not explicitly study the content of the collaborative programs, the process of topic definition differed between collaborations. In some cases, the industry partners specified the content to be covered in the courses. In other cases, either the university partners were responsible for this task or a mixed group, formed by university and industry representatives, specified the content.

Admission Procedure: The admission procedure used by the various collaborations also differed. In some collaborations the industry partner selected students, while in others students were required to fulfil the academic institution's program entrance requirements. There were also differences in how students elected to join the program. In some collaborations the in-

dustry partner dictated that certain employees must enter the program, whereas in other programs the course was offered to all the employees of certain departments and enrollment was voluntary for the employees who had a personal interest in taking the course.

Financial Models: The final main difference between the collaborations was the financial model used to support the program. In some collaborations the industry partner paid on a per student basis, while in others the industry partner provided the academic partner with a specific amount of funding, irrespective of the number of students. In other cases, the academic institution offered courses with open enrollment, which were taken by students from the industry partner.

In addition to these three dissimilarities, other minor differences between the programs included location, timing of class offerings, and content. It appears from our study that these dissimilar characteristics have no significant impact on the success or failure of a collaboration.

3.3 Benefits

In the process of evaluating the collaborations, we identified four benefits that resulted from the collaborative reeducation efforts:

Knowledge Enrichment of University Teachers: The faculty that participated in the collaborative program appeared to gain increased exposure to software development practices in the real world. This exposure allows practical knowledge to migrate into the curriculum and provides faculty with a utilitarian view of software development.

Self-Supporting Programs: The industry/university reeducation collaborations that we studied were entirely self-supporting and, in some cases, financially profitable for the academic partner. It appears that, while the university partner may have to make some up-front investments for things like course and curriculum development, the income from industry provides sufficient funding to support the program. It should also be noted that collaborations also provide some indirect economic benefits to the academic partners in the form of increased visibility of the university and its programs.

Knowledge Enrichment of Industry Employees: The expanded knowledge and skill set of employees is the primary benefit identified by industry partners in the collaborations that we studied. This enhanced skill set benefits the industry partner as the company is able to produce better quality software in a more efficient manner, while also benefiting the employees through the advancement of their professional careers.

More Competitive Software Industries: Lastly, industry/university reeducation collaborations result in a benefit to the software industry as a whole. Some companies reported that their clients looked upon their participation in the collaboration very favorably, resulting in increased prestige in the software community. This benefit appears to result from the improved technical expertise gained by the employees.

In our evaluation of the detailed questionnaires and industry opinion surveys, we also identified other indirect benefits obtained from the collaborative program. Industry feedback indicates employees are less likely to leave the company when they are participating in a collaborative reeducation program, and the opportunity to participate in a collaborative program motivates others to continue working at the partner company.

3.4 Industry Viewpoint

Since the industry partner in the industry/university reeducation program could be considered a main customer of the collaboration, we decided to survey those industry partners to ascertain their opinions on the success of the collaborations. This section discusses the industry opinions expressed in the industry opinion survey.

In general, the industry partners appeared to be very satisfied with the collaborations and their outcome. All of the industry partners indicated that they were either satisfied or very satisfied with the program development and participant selection. All industry partners thought that the program management and oversight, student enrollment procedure, and format, content, and pace of the program was acceptable. When asked about the classroom environment, the majority of the respondents rated both the instructors' knowledge and the facilities as outstanding, while most respondents rated the instructors' teaching skills sufficient. All of the respondents indicated that they felt that the number of participants in the program was the correct size. When queried about their opinions of the program results, all of the industry partners indicated that they felt that the program was successful and that they gained knowledgeable, competent employees.

When asked about areas of improvement, some industry partners stated that incorporating more domain knowledge into the program could strengthen program content. However, few companies also indicated that they needed to provide employees with more flexibility in incorporating the program into their work environment. This finding appears to indicate that industry desires more relevant software engineering education.

One interesting feature that we noted in our evaluation of the industry opinion surveys was that only three of the industry partners track their employees who have completed the program. The tracking ranges from monitoring the employees' progress to a follow-up with the

employee after program completion. Progress is monitored in terms of job promotions and to determine if employees are using the skill set acquired in the collaborative program.

Lastly, industry partners were asked about the important lessons learned from participating in the collaboration. The need for pre-planning before implementing a collaborative program was identified as a significant lesson by most industry partners. The need to clearly discuss and define requirements and expectations was identified as another important lesson gained.

In addition to the common and dissimilar characteristics of collaborations, we have also noticed that the main reason why collaborations appear to cease to exist is management turnover and the corresponding change of priorities.

4 Evaluating Knowledge/Skill Transfer

Having identified a set of features common across industry/university reeducation collaborations, the next step in understanding the practical benefits of the collaborations was to investigate the transfer of knowledge and skills that occurs within them. We used the alumni survey to elude the opinions of alumni on topics that would enable us to assess the efficacy and efficiency of knowledge transfer. In this section, we describe this effort and the results. Note that the evaluation process we have followed could be used for any software engineering program, either graduate and undergraduate, to analyze its efficacy in terms of practical usefulness and its efficiency in terms of the effort spent during the teaching process and the level of knowledge acquired by the students.

4.1 Alumni Survey Description

Alumni of four different collaborations were surveyed to analyze knowledge transfer in their collaborations. Three of the collaborations were taught in person and one was taught using distance education. The alumni of the four programs are software development practitioners with a variable number of years of experience. Each of these four collaborations targets between 15 and 25 students per year.

A total of 31 surveys for the four different collaborations (11, 10, 7, and 3 responses) were collected. Two collaborations were located in Spain (Madrid) and two in the United States (Georgia, Pennsylvania). Although the number of surveys is insufficient for a formal statistical analysis, the survey results allow us to make some general observations about the efficacy of the collaborative programs.

The alumni survey contained two parts. In the first part, the alumni were questioned about the major Software Engineering Body of Knowledge (SWEBOK) topics [Hilburn 99]:

- software requirements
- software design
- software construction
- software testing
- software maintenance
- software configuration management

- software engineering management
- software engineering process
- software engineering tools and methods
- software quality

The alumni were asked to evaluate how much each of these subjects was emphasized, how useful the knowledge acquired in each subject was for their current tasks, and whether they acquired sufficient knowledge of each subject. They used the assessment criteria shown in Table 1.

Table 1: Subject Assessment Criteria Concerning Emphasis, Usefulness, and Alumni's Knowledge

Emphasis	Wide Coverage with Examples	Moderate Coverage	Theory Only	Brief Coverage	Not Taught
Usefulness	Essential	Very Useful	Moderately Useful	Not Useful	Not Taught
Knowledge	Thorough Understanding (Expert)	Moderate Understanding	Understanding of Concepts	Introduction to the Topic	Not Taught

The second part of the survey asked alumni to assess the program's effect on their educational and professional status by indicating the extent to which they agreed (strongly agree, agree, disagree, and strongly disagree) with the following statements:

- Overall, my software engineering education has adequately prepared me for my current job.
- My education provided me with knowledge and skills that contribute to my job performance.
- My education has improved my software development skills.
- My education has improved my communication skills.
- My education has improved my technical skills.
- My education has improved my organizational skills.
- My education was highly relevant to state-of-the-art practices in my field.
- My education provided me with a useful reputation as a software engineer.
- My education contributed to the development of my professional career.
- I would recommend participation in my educational program to others.

The results of the alumni survey are described in the next section.

4.2 Alumni Survey Results

To understand how the programs affected the alumni's jobs, we surveyed them to determine the software engineering activities performed in their current positions. The results are shown in Table 2. (The percentage column does not add up to 100, as some alumni performed more than one activity type.)

Table 2: Current Alumni Job Types

Current Job Type	Percentage of alumni
Requirements	36%
Design	25%
Coding	14%
Test Engineer	29%
Configuration Management	11%
Maintenance	25%
Documentation	21%
Quality Assurance	11%
Project Management	50%

4.2.1 Results of the Content Measurement

Figure 2 shows the results of all 31 surveys regarding the emphasis, usefulness, and sufficiency of knowledge conveyed for each of the SWEBOK topics.

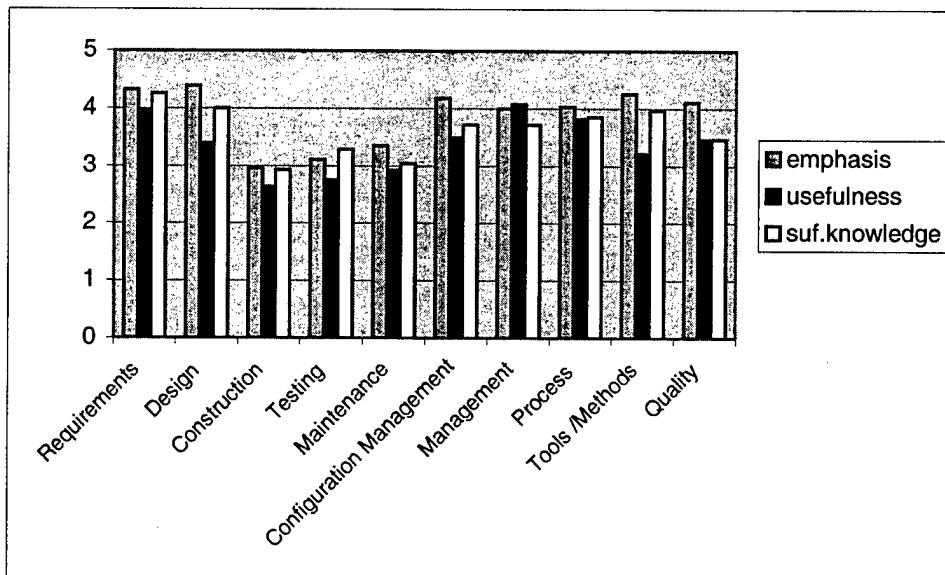


Figure 2: Weighted Values of Emphasis, Usefulness, and Sufficiency of Knowledge for Each Subject

The subjects analyzed are shown along the x -axis and the total assessment of the emphasis, usefulness, and sufficiency of knowledge acquired are shown along the y -axis. Therefore, the greater the value on the y -axis, the higher alumni considered the emphasis, usefulness, or sufficiency of knowledge for that subject.

Overall, practitioners who participated in the collaborative programs have a positive impression of the effectiveness and usefulness of the skills that they gained. These results differ somewhat from Lethbridge's study on the applicability of software engineering knowledge [Lethbridge 01], which indicates a larger gap between the transferred knowledge and its usefulness in industry. One reason for this difference could be the fact that the majority of the collaborative efforts we investigated involved postgraduate (master's level) education, which is typically more focused than undergraduate education. This is to be expected, particularly in the case of the programs in question, where the industrial partners have a considerable influence on the definition of the program contents.

Figure 2 indicates that the alumni consider the knowledge acquired about software management and software requirements most useful, whereas the knowledge related to software construction, software testing, and software maintenance is somewhat less useful. Since a higher percentage of alumni perform project and requirements management tasks, as shown in Table 2, this is not an unexpected result. Also, alumni admitted that, before starting the course, they had little knowledge of these subjects. The results for the topics of testing and maintenance indicate that, while the percentage of alumni who perform these tasks is not among the lowest, alumni do not consider the knowledge acquired to be very useful. The reason given by the alumni was that they already had knowledge of these subjects before they started the course and, therefore, the usefulness of the new knowledge received was not as high as in software management and requirements.

The ideal result would be for the y -axis values for all three measures (emphasis, usefulness, and sufficiency) to be equal for each subject, although they would not be the same for all the subjects, as the usefulness of each subject is different. A quantitative analysis of the chart in Figure 2 shows that the average pair-wise correlation for the three characteristics ranges from 78%, between emphasis and usefulness, through 79%, between usefulness and sufficiency of knowledge, to 89%, between emphasis and sufficiency of knowledge. This means that the levels of the three characteristics are quite related, although far from the ideal 100% value.

Usefulness and Sufficiency of Knowledge

The relationship between usefulness and sufficiency of knowledge indicates the effectiveness of the programs in terms of the usefulness of the knowledge transferred for the current development tasks of the alumni. The closer the rating of these two factors, the more effective the alumni viewed the program. As shown in Figure 2, except for the software management sub-

ject, the levels of sufficiency of knowledge for the different subjects are greater than or equal to the usefulness values, indicating that the programs are viewed as being very effective.

In the case of software management, the emphasis is also slightly lower than the usefulness rating, which may indicate a need for programs to reexamine content related to software management, as well as to review the process by which this topic is taught. This is a key point, because, as shown in Table 2, half the alumni surveyed perform tasks related to this topic. Note that although the emphasis with which this discipline is taught is greater than the sufficiency of knowledge, the emphasis is also slightly lower than usefulness. Analyzing the data that give rise to this situation in depth, we find that, although there are no statistically significant differences between the data of the three collaborations, it is qualitatively appreciable that the distance program data have lower levels of sufficiency of knowledge acquired on this subject. This means that special attention should be paid to the teaching methods used in distance training, by encouraging, for example, more interaction with students to raise the level of sufficiency of knowledge. As we shall see later, this is not the only distance program subject where differences in alumni responses are apparent (although they are mostly not statistically significant), which means that the intuitive idea that distance training has more handicaps than classroom education and, therefore, calls for more teaching preparation effort can be corroborated to some extent.

Overall, the 79% correlation between usefulness and sufficiency of knowledge shows that alumni are satisfied with the knowledge gained from the collaboration. A large difference between the sufficiency of knowledge rating and a greater usefulness rating indicates that programs are providing more knowledge or skills than required by the alumni. As shown in Figure 2, topics that exhibit this characteristic include software tools/methods and software design. Since only 25% of alumni perform design tasks, as shown in Table 2, alumni may benefit from education at a higher level of abstraction.

Emphasis and Sufficiency of Knowledge

The relationship between emphasis and sufficiency of knowledge is important mainly from the viewpoint of the teachers, as it indicates how efficient their teaching is. So, the closer the levels of these two characteristics are, the more efficient the teaching would be, and the higher the level of sufficiency with respect to emphasis the better. As mentioned above, the correlation value between the two characteristics is 89%, which indicates that it is generally true that if a lot of emphasis is placed on the subject taught, the level of knowledge acquired by the students on the subject is also high. However, it is interesting to analyze which subjects have bigger or smaller differences. The biggest differences are found in the subjects of software quality and software configuration management, which means that the sufficiency of the knowledge acquired by the students on these subjects is removed from the emphasis with which these subjects are taught (although it is appropriate in relation to their practical usefulness, as shown in Figure 2). There is more than one possible explanation for this. One is that the alumni had little previous knowledge of these subjects, as they themselves admit-

ted. Alternatively, the teaching methods used might not be the most suitable. For example, in two collaborations, the teaching was based on real quality and configuration management plans, which meant that students had to make a tremendous effort to try to understand these complex concepts quickly. It would be advisable to adapt the process by which these disciplines are taught to deal with the complexity of the subjects gradually, first using fictitious and simple quality and configuration management plans and gradually raising their complexity. Accordingly, the students would be able to internalize the complexity of these disciplines progressively, and the match between the levels of sufficiency of knowledge and emphasis placed on the teaching would tend to improve. Special attention should be paid to distance education, as, again, it was found that the difference between these characteristics in the results of the distance program surveys was bigger than in the other surveys, although it was not statistically significant.

The subjects in which the levels of both emphasis and sufficiency of knowledge are closer are also worth mentioning, because, as we said above, this would be an indication that the teaching process followed is efficient. These subjects are software construction and software requirements. An examination of their teaching processes revealed that, generally, those processes are based on practice at different levels of complexity in all programs.

Finally, there remains the particular case of software testing, where the sufficiency of knowledge is above the level of emphasis. A direct interpretation of these data could be that the efficiency of the teaching process of this knowledge is exceptionally good. However, a more detailed qualitative analysis of the alumni's responses and their justifications shows that they already had informal background knowledge of the subject and that, although the teaching process was based on theoretical explanations or simple examples of the different testing techniques, they could use these simple explanations to sort out and formalize the informal knowledge that they already had. Therefore, the value of sufficiency of knowledge was higher than the emphasis value.

Emphasis and Usefulness

We will not go into the relationship between the emphasis with which the knowledge is taught and its usefulness in detail, as, although this relationship can provide a preliminary impression of the efficiency of the teaching process for the current activities of the attendants, the real indicator of this measure is the relationship between usefulness and sufficiency of knowledge discussed above. However, it should be highly correlated (as indeed it is, at 78%), and the level of emphasis should be higher than the level of usefulness of the knowledge (as shown in Figure 2, except for project management as discussed above).

4.2.2 Results of the Subjective Assessment

As mentioned above, the survey also included a subjective assessment of the programs with regard to their effect on both the educational and professional status of the alumni. Table 3 shows how the alumni responded to each statement in the assessment.

Table 3: Percentage of Alumni Who Evaluated Each Statement

Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
My education has improved my software development skills.	25%	54%	21%	0%	0%
My education has improved my communication skills.	18%	43%	32%	7%	0%
My education has improved my technical skills.	18%	71%	11%	0%	0%
My education has improved my organizational skills.	21%	54%	21%	4%	0%
My education was highly relevant to state-of-the-art practices in my field.	32%	36%	25%	7%	0%
I am confident in my ability to apply the knowledge and skills that I learned in my educational program.	29%	36%	32%	4%	0%
There is support from management for me to employ the knowledge and skills that I learned in my education program.	18%	36%	21%	11%	14%
I am motivated to employ the knowledge and skills that I learned in my education program to my job.	43%	46%	11%	0%	0%
My education provided me with a useful reputation as a software engineer.	7%	21%	36%	21%	14%
My education contributed to the development of my professional career.	11%	46%	21%	14%	7%
I would recommend participation in my educational program to others.	46%	43%	11%	0%	0%

These results are shown graphically in Figure 3, where the x -axis shows the opinion statements and the y -axis shows the alumni's assessment expressed as a percentage. These results indicate that the alumni generally agree that the programs improved their skills, with 70% of the alumni agreeing that the programs improved their technical skills. This percentage is slightly lower (although this difference is not statistically significant) for the distance program.

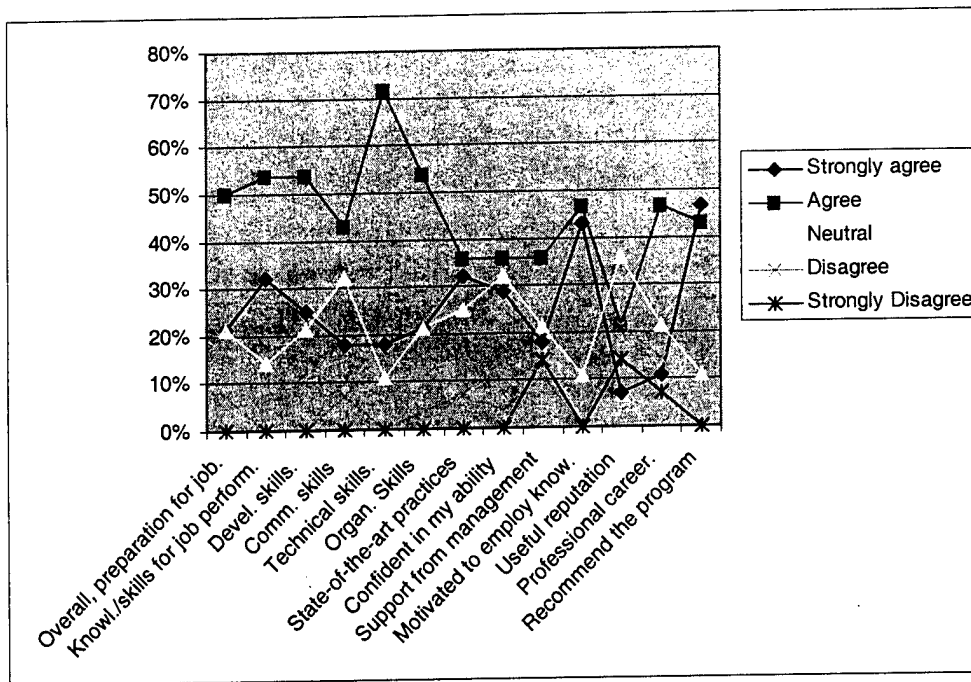


Figure 3: Graphical Representation of Attendants' Assessment of the Program Effect

Regarding management support, it is interesting to note that, although a majority of the alumni state that this support exists, about 25% of them noted that they received no such assistance (disagree or strongly disagree). Since it is the industrial partners that actually finance and promote the programs in these collaborations, it is somewhat contradictory that there are alumni who do not find the support to apply the acquired knowledge. In this respect, it would be useful to strengthen management support.

Also, the opinion of the alumni with regard to the improvement of their reputation as software engineers is noteworthy, where the percentage of alumni who either disagree or strongly disagree is higher than those who agree or strongly agree. It is perhaps a little early to interpret this information, as is the case with the question regarding the contribution of the program to the development of the professional career of the alumni. Although the majority agree with the latter statement, over 20% either disagree or strongly disagree. Since the respondents are alumni who completed the latest programs, a correct assessment of these statements may not be possible until after some time has passed.

Finally, it should be mentioned that almost 90% of alumni would recommend the program they took to colleagues. On this point, it is noteworthy that there was a significant difference between the distance program and classroom program alumni. The percentage of alumni who strongly agree with this statement was significantly higher for the classroom programs as compared with the distance program, whereas the percentage of distance alumni who agree with the statement was significantly higher than that of alumni in the classroom programs

who agree. Again there is an underlying intuitive idea that distance programs may be less appealing to students. However, many of the alumni participating in the distance program recognized the usefulness of distance education, as they stated that, because of time and mobility limitations, they would have been unable to take the program if it had required class attendance.

5 Conclusion

In the course of the two-year investigation into industry/university collaborations to reeducate non-software personnel to become software engineers, the I/U subgroup of the WGSEET has compiled several useful observations about such collaborations. The primary observation is that these collaborations provide significant benefits for all three major stakeholders: the industry partner, the academic partner, and the students. Industry feedback indicates that companies benefit from a larger, more highly trained workforce with a higher level of knowledge and skills, while retaining large stores of domain knowledge held by the existing employees. Universities benefit from increased visibility of their programs, as well as from the dissemination of a real-world vision of software engineering throughout the faculty. Students benefit by becoming more highly skilled workers, with higher job satisfaction and more career mobility.

In addition to the benefits already being reaped from industry/university reeducation collaborations, we suggest that further opportunities for profit exist. The interactions between industry and universities provide fertile ground for applied research that combines the real-world knowledge of the company with the theoretical background of the university to produce solutions to practical problems. We also feel that academic institutions must maintain close ties to the industry that employs their graduates.

One area of concern that we identified in our investigation was the apparent relatively short life of industry/university collaborations for reeducating software engineers. Given the amount of effort required to construct such a program and the continuing shortfall of qualified software engineers, we believe that maintaining these programs is of value to the partners involved and to the larger discipline of software engineering.

References

- [Beckman 99] Beckman, K. *Directory of Industry and University Collaborations with a Focus on Software Engineering Education and Training* (CMU/SEI-99-SR-001, ADA360577). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999.
<<http://www.sei.cmu.edu/publications/documents/99.reports/99sr001/99sr001abstract.html>>.
- [Ellis 01a] Ellis, H. J. C.; Mead, N. R.; Moreno, A.; & MacNeil, P. "Can Industry and Academia Collaborate to Meet the Need for Software Engineers?" *Cutter IT Journal* 14, 6 (June 2001): 32-39.
- [Ellis 01b] Ellis, H. J. C. "Summary of the Initial Results of the University/Industry Survey Performed by the Industry/University Subgroup of the Working Group on Software Engineering Education and Training." *Forum for Advancing Software Engineering Education* 11, 1 (Jan. 2001).
- [Ellis 02] Ellis, H. J. C.; Mead, N. R.; Moreno, A.; Tanner, C. D.; & Ramsey, D. "Characteristics of Successful Collaborations to Produce Educated Software Engineering Professionals." *Computer Science Education* 12, 1-2 (Spring 2002): 119-140.
- [Frailey 98] Frailey, D. & Moore, F. "Maintaining a Capable Software Engineering Pool." *Proceedings of the 1998 Software Technology Conference*. Salt Lake City, Utah, Apr. 19-23, 1998. Salt Lake City: Utah State University, 1998.
- [Frezza 99] Frezza, S. T. & Hazen, S. L. "Integrating an Industrial Practicum into a Graduate Embedded Software Engineering Program." *Proceedings of the Frontiers in Education Conference*. San Juan, Puerto Rico, Nov. 10-13, 1999. New York: IEEE Computer Society Press, 1999.

- [Hazen 97] Hazen, S. L. & Frezza, S. T. "An Integrated Embedded Software Engineering Program and Practicum." *Proceedings of the Frontiers in Education Conference*. Pittsburgh, Pennsylvania, Nov. 5–8, 1997. New York: IEEE Computer Society Press, 1997.
- [Heiman 01] Heiman, R. V.; Byron, D.; Mason, R. P.; & Marks, M. *Worldwide Software Market Forecast Summary, 2001-2005*. Framingham, MA: IDC, Sept. 2001 (Doc. #25569).
- [Hilburn 99] Hilburn, T., et al. *A Software Engineering Body of Knowledge, Version 1.0* (CMU/SEI-99-TR-004). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999.
<<http://www.sei.cmu.edu/publications/documents/99.reports/99tr004/99tr004title.html>>.
- [Kellinson 00] Kellinson, J. W. & Tate, P. "The 1998-2008 Job Outlook in Brief." *Occupational Outlook Quarterly* 44, 1 (Spring 2000): 2–39.
- [Lethbridge 01] Lethbridge, T. "What Knowledge is Important to a Software Engineer?" *Computer* 33, 5 (May 2001): 44–50.
- [Mead 99] Mead, N. R., et al. "Industry/University Collaborations: Different Perspectives Heighten Mutual Opportunities." *Journal of Systems and Software* 49 (1999): 2–3.
- [Meyer 01] Meyer, B. "Software Engineering in the Academy." *Computer* 34, 5 (May 2001): 28–35.
- [Mingas 99] Mingas, C., et al. "How We Teach Software Engineering." *Journal of Object-Oriented Programming* (February, 1999).
- [Modesitt 01] Modesitt, K.; Bagert, D.; & Werth, L. "International Academic Software Engineering: Results of the First Annual Survey." *Proceedings of the 19th IASTED International Conference on Applied Informatics*. Innsbruck, Austria, Feb. 19–22, 2001. Anaheim: ACTA Press, 2001.

[Wholin 99]

Wholin, C. & Regnell, B. "Achieving Industrial Relevance in Software Engineering Education." *Proceedings of the 12th Conference on Software Engineering Education and Training*. New Orleans, LA, Mar. 22–24, 1999. New York: IEEE Computer Society Press, 1999.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE July 2002	3. REPORT TYPE AND DATES COVERED Final		
4. TITLE AND SUBTITLE Reeducation to Expand the Software Engineering Workforce: Successful Industry/University Collaborations		5. FUNDING NUMBERS F19628-00-C-0003		
6. AUTHOR(S) Heidi J. C. Ellis, Ana Moreno, Nancy R. Mead, Stephen B. Seidman				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2002-SR-001		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		12B DISTRIBUTION CODE		
13. ABSTRACT (MAXIMUM 200 WORDS) Software produced worldwide is growing at a phenomenal rate as software is used in such diverse products as automobiles, homes, and airplanes. In addition, the increasingly global business climate and expanding emphasis on distributed computing have accelerated the need for business software. However, there is currently an inadequate number of software engineers to produce and maintain software to meet this demand. One possible solution to correcting this shortfall is reeducating existing non-software engineering employees to become software engineers. For the past two years, the Industry/University (I/U) subgroup of the Working Group on Software Engineering Education and Training has been investigating active collaborations between companies and universities in which non-software professionals and practitioners who lack formal software education are reeducated to become software engineers. This paper reports on the I/U subgroup's findings by describing their approach to the investigation, the factors involved in successful collaboration construction and execution, and alumni views of the knowledge and skills transferred by the collaboration.				
14. SUBJECT TERMS software engineering education, software engineering training, software engineers, reeducation, industry/university collaborations		15. NUMBER OF PAGES 39		
16. PRICE CODE				
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	